BÖLÜM 11

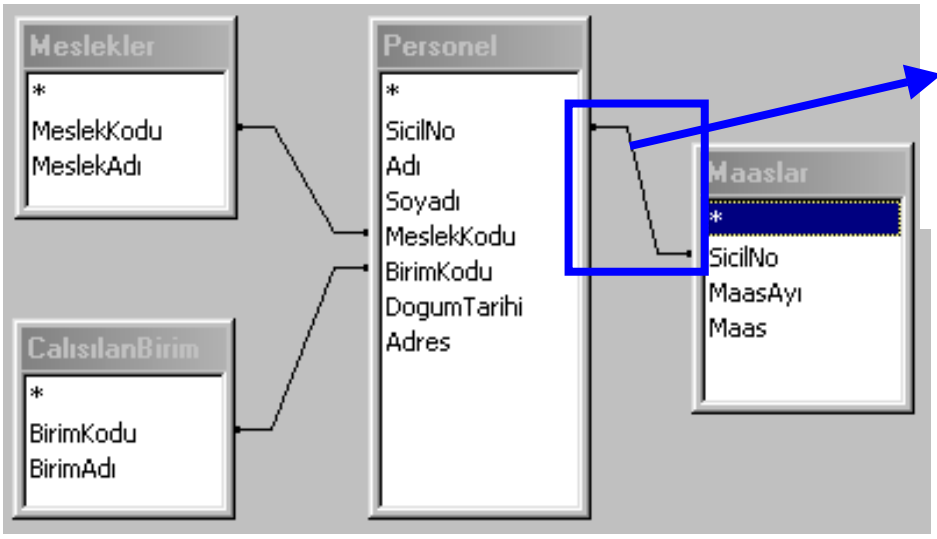
11. SQL 'de JOIN (BİRLEŞTİRME) İŞLEMİ

11.1. JOIN (Birleştirme) İşlemi

Veri tabanı kayıtları oluşturulurken bütün bilgiler bir tabloda değil de, birkaç tablo üzerinde tutulur. Bu dataların daha düzenli olmasını, gereksiz veri tekrarlarının engellenmesini ve veri yönetimini kolaylaştırır. Daha önce bu konu ile ilgili 4. ve 5.Bölümde veri tabanı ve normalizasyon konusundan detayları ile bahsedilmiş ve örnekler verilmiştir.

Bu bölüme kadar yazılan tüm sorgularda tek tablo üzerinde işlem yapıldı. Bu bölümde, ilişkisel veri tabanlarının ve SQL'in çok önemli ve yararlı bir özelliği olan birleştirme ("join") işlemi anlatılacaktır. İlişkisel veri tabanlarına "ilişkisel" denmesinin nedeni olan Birleştirme işlemi iki veya daha çok tablo arasında bağlantı ("link") kurarak tek bir tablo oluşturur ve sorgu bu tablo üzerinde çalışır .

Birden fazla tablo üzerinde işlem yapılacağı zaman; select cümleciğinden sonra tabloadı.tablo_alanları aralarına virgül konularak yazılır ve kullanılacak tablolar from cümleciğinden sonra aralarına virgül konularak yazılır. Daha sonra da WHERE şartı ile tablolar arasındaki ilişkili ortak alanlar eşitlenir. Buradaki en önemli özelliğin iki tablo arasında ilişkili ortak bir alanın olması gerektiğidir. Aşağıda projemize ait tablolar ve ilişkili alanlar belirtilmiştir.

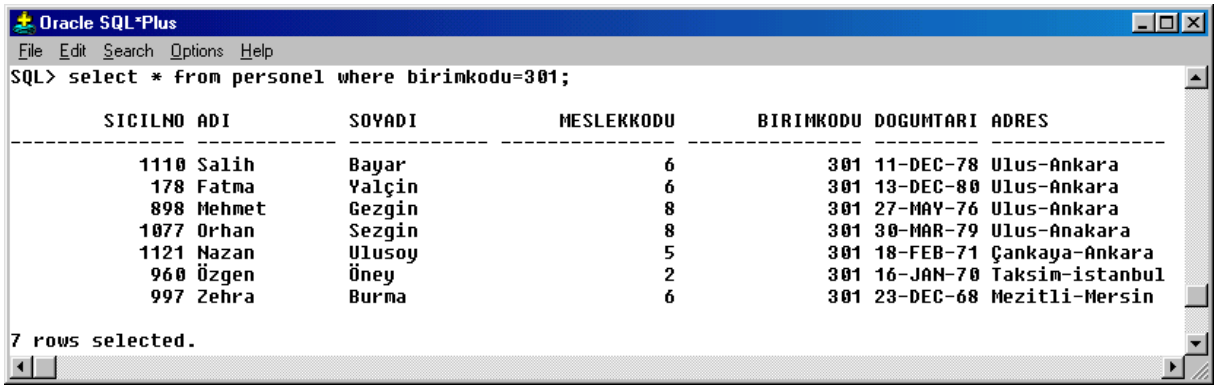


Şekil 11.1.1. Örnek Personel Projenin MS Access deki diyagramı

Tablo İsimleri	Ortak Alanlar
Personel-Maaş tablosu	SicilNo
Personel-Meslekler tablosu	MeslekKodu
Personel-CalısılanBirim tablosu	BirimKodu

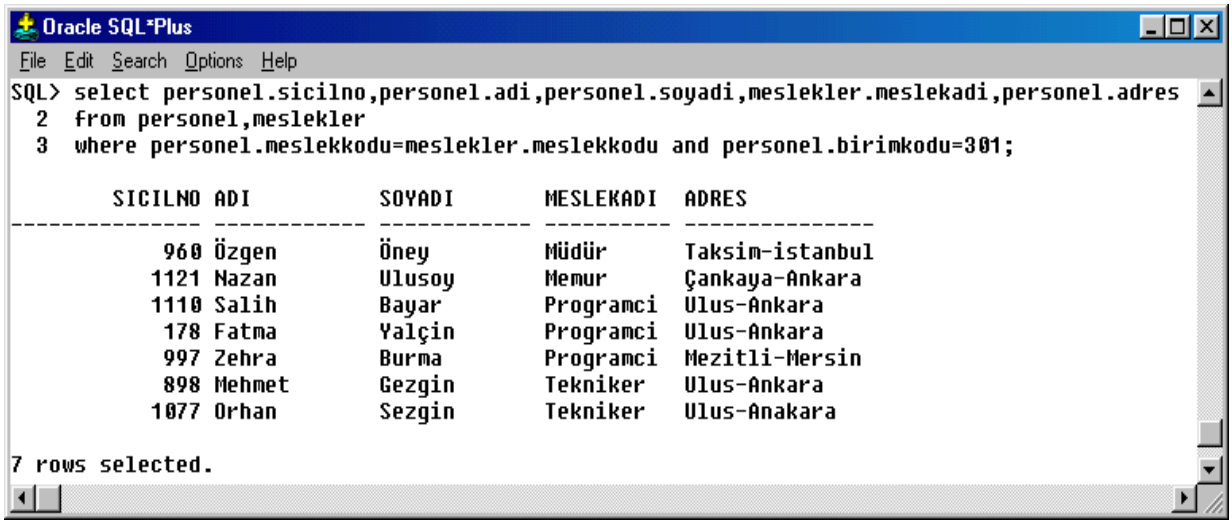
Projedeki her tablonun bir alanı ana tablo olan personel tablosunun bir alanı ile ortaktır ve tanımlanan alanlar aynı özelliktedir.

Örnek-1) ***SQL> select * from personel where birimkodu=301;***
BirimKodu 301 olan personelin listesini verir.



Şekil 11.1.2. Join Sorgu Ekranı-1

Bu sorgu sonucunda personele ait birimler birimKodu 301 olarak ve mesleklerde meslekkodu olarak listelenmektedir. Kod olarak listelenen bilgiler çok açıklayıcı değildirler. Örneğin Salih Başar isimli personelin meslekkodu 6 olarak görüntülenmekte ama 6 kodunun hangi meslek olduğu anlaşılamamaktadır. Personel tablosundaki meslekkodunun karşılığı olan değer meslekler tablosundan bulunarak getirilmelidir. Bu meslekkodlarının hangi mesleğe denk geldiğini bulmak için personel ve meslekler tablosu arasında join yapılmalıdır. Şöyleki; personel tablosunda Salih Bayar isimli personele ait 6 olan meslek kodu meslekler tablosunda aranacak ve meslekler tablosunda meslekkodu 6 olan kayıtta bulunan programcı mesleği personelin mesleği olarak yazılacaktır. Bu işlemden sonra meslek alanına programcı yazılacak ve veriler daha anlamlı hale gelecektir. Bu karşılaştırma işlemini yapan ifade WHERE karşılaştırmasından sonra kullanılacak olan ifadedir.



Şekil 11.1.2. Join Sorgu Ekranı-2

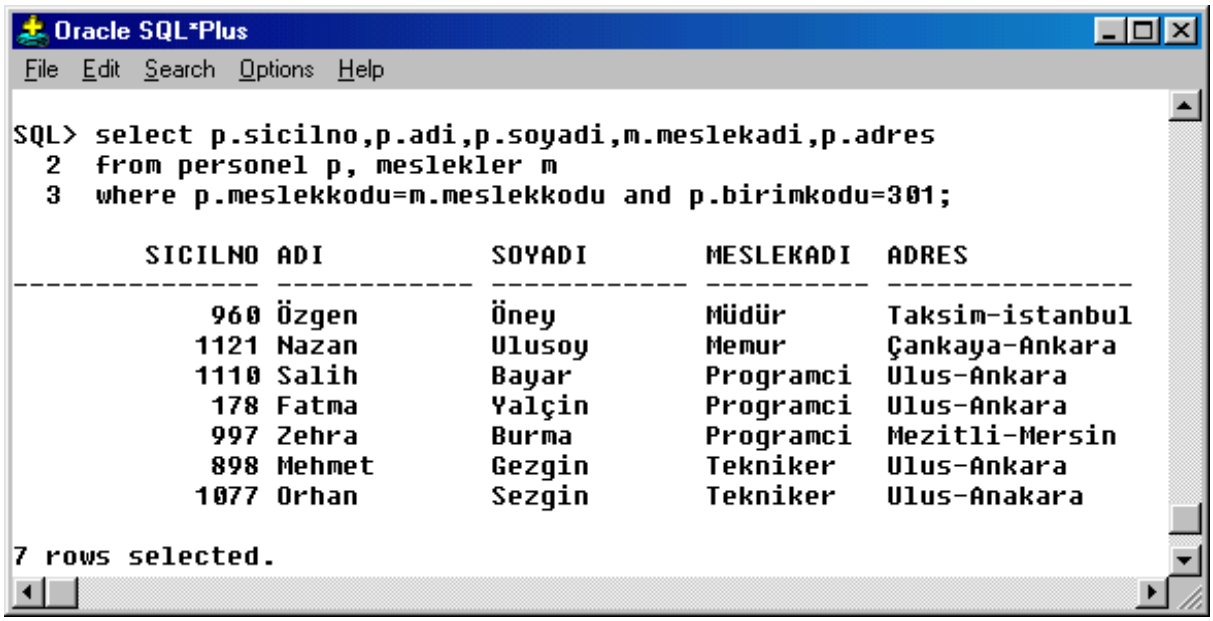
Sorgu çok uzun ve karışık gibi görünmesine rağmen aslında çok basittir. Select komutunun ardından birden fazla tablo kullanıldığı için alanın hangi tabloya ait olduğunu bildirmek için listelenecek her alan için ***tabloadı.tablo_alanı*** kullanılmıştır. ***Personel.sicilno*** veya ***meslekler.meslekkodu*** gibi. From komutundan sonra kullanılan tablolar virgül ile ayrılarak yazılmıştır. Daha sonra personel tablosundaki meslekkodunun meslekler tablosundaki karşılığını bulmak için

where personel.meslekkodu=meslekler.meslekkodu karşılaştırması yapılmıştır.

and personel.birimkodu=301 ifadesi tüm personel yerine 301 nolu birimde çalışan personelin listelemesi içindir.

and And ile iki şart birbiri ile birleştirilmiştir.

Yukarıdaki sorgu aşağıdaki gibi de yazılabilir.



Şekil 11.1.3. Join Sorgu Ekranı-3

İki sorguda aynıdır. Tek fark yazılım farkıdır. Dikkat edilirse p.sicilno gibi ifadeler vardır. Burada olduğu gibi “from” sözcüğünden sonra yazılan tablo adları için bir boşluk bıraktıktan sonra kısa bir isim verilebilir. Böylece uzun tablo adını sürekli yazmaktansa o tablo adı için “alias” olarak adlandırılan kısa isim kullanılabilir. Alias olarak verilen tablo ismi alanlar listelenirken yazım olarak büyük kolaylık sağlamaktadır. Burada personel tablosu için p, meslekler tablosu için m kullanılmıştır. En çok kullanılan yöntem ilk tabloya a harfinden başlamak, diğerlerine de b,c,… gibi alias isimler vermektir.

NOT : where personel.meslekkodu=meslekler.meslekkodu Bu ifade kodun karşılığını bulmak açısından en önemli olan ifadedir. kullanılmaz ise sorgu sonucu yanlış çıkacaktır. Şöyleki; her bir personel meslek tablosundaki tüm mesleklere karşılık gelecek şekilde listelenecektir.

Oracle SQL*Plus

File Edit Search Options Help

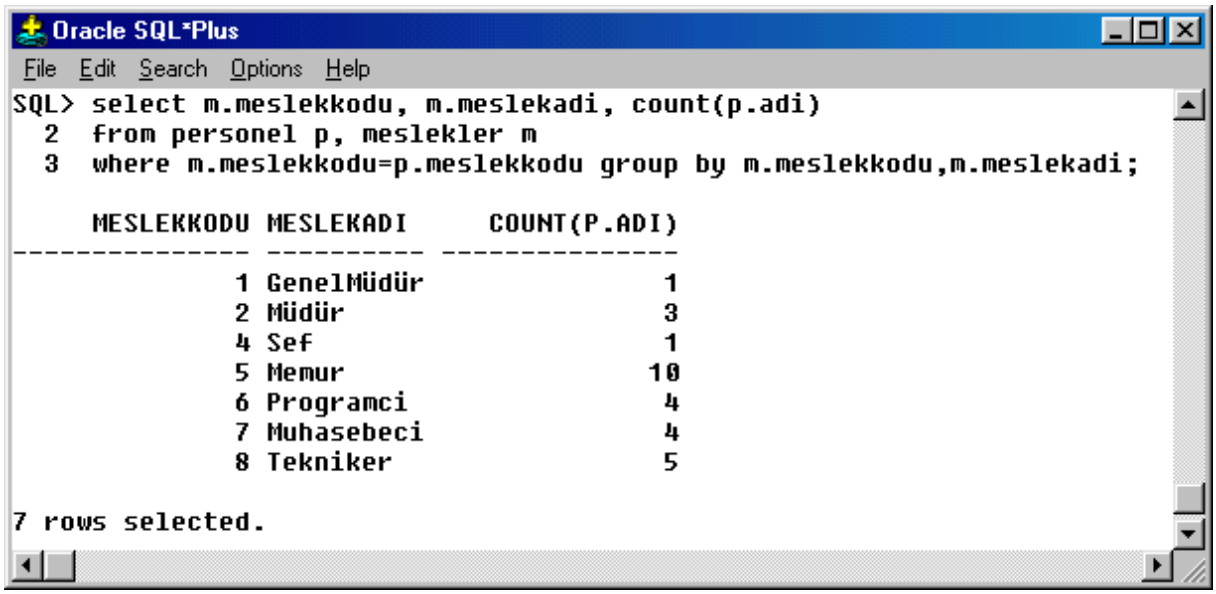
```
SQL>  select p.sicilno,p.adi,p.soyadi,m.meslekadi,p.adres
  2  from personel p, meslekler m
  3   where p.birimkodu=301;

       SICILNO ADI        SOYADI      MESLEKADI  ADRES
-------------- ---------- ----------- ---------- ---------------
          1110 Salih      Bayar       Müdür      Ulus-Ankara
           178 Fatma      Yalçin      Müdür      Ulus-Ankara
           898 Mehmet     Gezgin      Müdür      Ulus-Ankara
          1077 Orhan      Sezgin      Müdür      Ulus-Anakara
          1121 Nazan      Ulusoy      Müdür      Çankaya-Ankara
           960 Özgen      Öney        Müdür      Taksim-istanbul
           997 Zehra      Burma       Müdür      Mezitli-Mersin
          1110 Salih      Bayar       Sef        Ulus-Ankara
           178 Fatma      Yalçin      Sef        Ulus-Ankara
           898 Mehmet     Gezgin      Sef        Ulus-Ankara
          1077 Orhan      Sezgin      Sef        Ulus-Anakara
          1121 Nazan      Ulusoy      Sef        Çankaya-Ankara
           960 Özgen      Öney        Sef        Taksim-istanbul
           997 Zehra      Burma       Sef        Mezitli-Mersin
          1110 Salih      Bayar       Memur      Ulus-Ankara
           178 Fatma      Yalçin      Memur      Ulus-Ankara
           898 Mehmet     Gezgin      Memur      Ulus-Ankara
          1077 Orhan      Sezgin      Memur      Ulus-Anakara
          1121 Nazan      Ulusoy      Memur      Çankaya-Ankara
           960 Özgen      Öney        Memur      Taksim-istanbul
           997 Zehra      Burma       Memur      Mezitli-Mersin
          1110 Salih      Bayar       Programci  Ulus-Ankara
           178 Fatma      Yalçin      Programci  Ulus-Ankara
           898 Mehmet     Gezgin      Programci  Ulus-Ankara
          1077 Orhan      Sezgin      Programci  Ulus-Anakara
          1121 Nazan      Ulusoy      Programci  Çankaya-Ankara
           960 Özgen      Öney        Programci  Taksim-istanbul
           997 Zehra      Burma       Programci  Mezitli-Mersin
          1110 Salih      Bayar       Muhasebeci Ulus-Ankara
           178 Fatma      Yalçin      Muhasebeci Ulus-Ankara
           898 Mehmet     Gezgin      Muhasebeci Ulus-Ankara
          1077 Orhan      Sezgin      Muhasebeci Ulus-Anakara
          1121 Nazan      Ulusoy      Muhasebeci Çankaya-Ankara
           960 Özgen      Öney        Muhasebeci Taksim-istanbul
           997 Zehra      Burma       Muhasebeci Mezitli-Mersin
```

Şekil 11.1.4. Join Sorgu Ekranı-4

Yukarıda where şartında eksik yazılan sorgu ekranının bir kısmı bulunmaktadır. Listelenen kayıt sayısı 56 dır. Bu sayı 7 (meslek sayısı) * 8 (301 de bulunan personel sayısı) çarpımıdır. Bu sorgu çalışan 30 personel ait bir liste olsaydı ve ***where personel.meslekkodu=meslekler.meslekkodu*** şartı yazılmasaydı liste 7*30 = 210 tane olurdu. Bu nedenle Where koşulunda ilişkili ortak alanların karşılaştırılmasına çok dikkat edilmelidir.

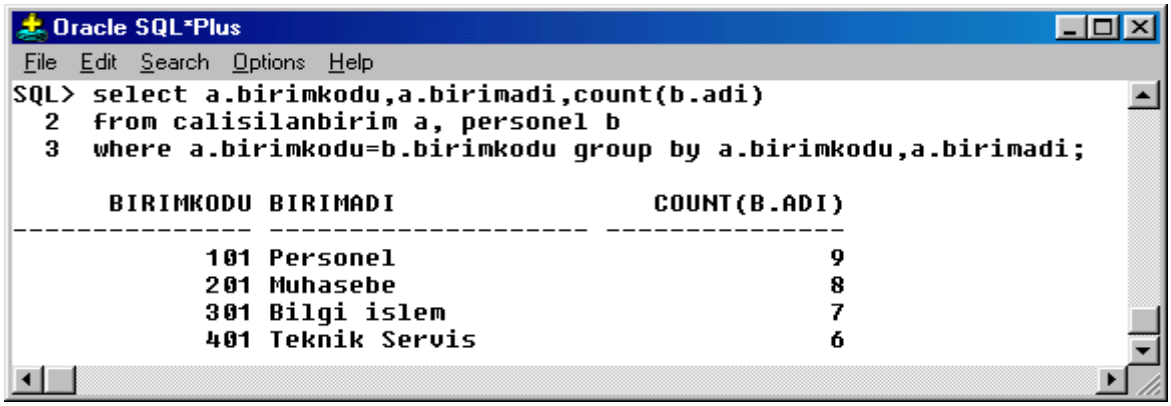
11.2. JOIN İşlemine Ait Örnekler

Örnek-1) Personel tablosundaki her meslek grubunda kaç personel olduğunun listesini veren sorgu.



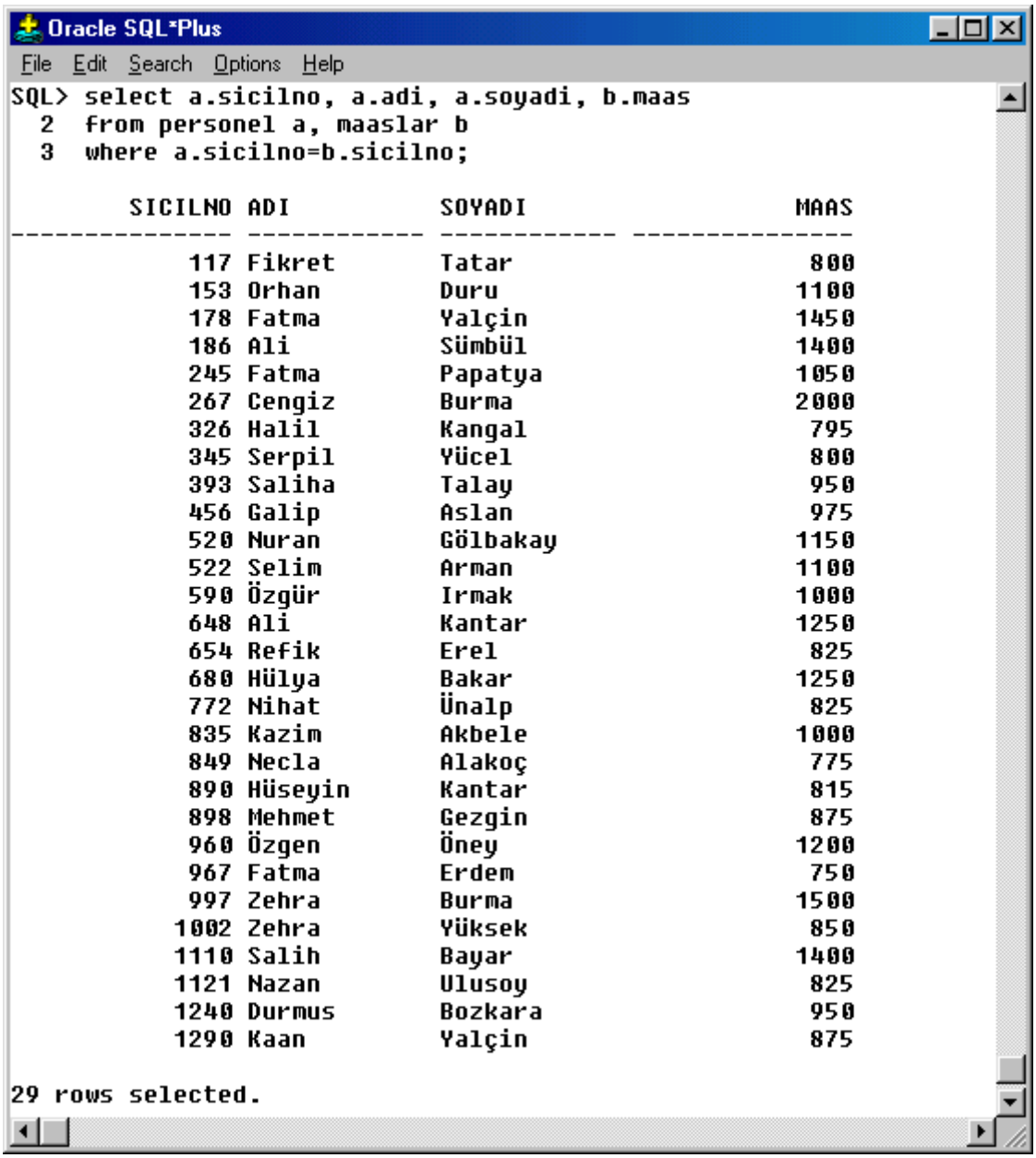
Şekil 11.2.1. Örnek Join Sorgu Ekranı-1

Örnek-2) Personel tablosundaki her birimde çalışan kaç personel olduğunun listesini veren sorgu.



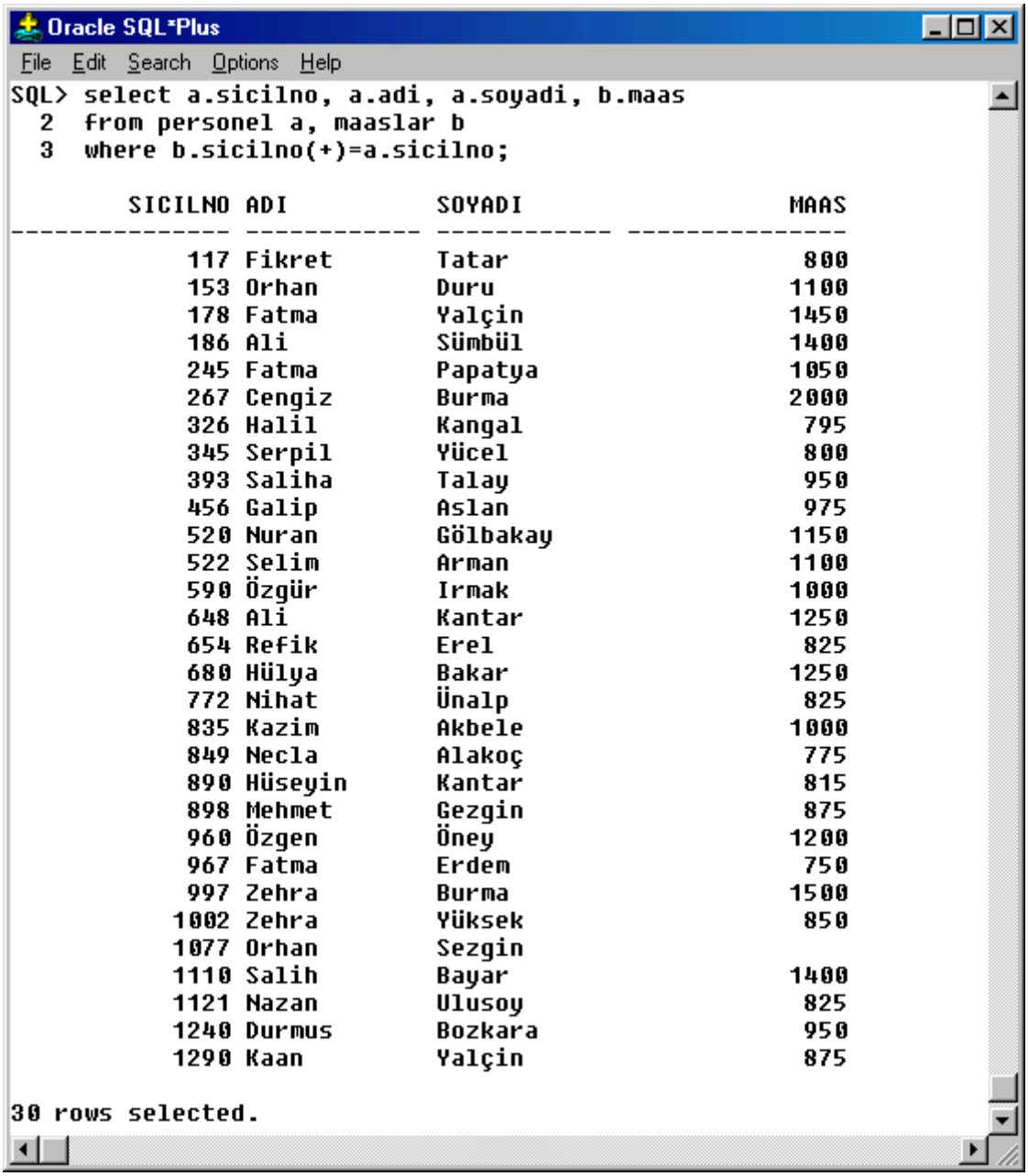
Şekil 11.2.2. Örnek Join Sorgu Ekranı-2

Örnek-3) Tüm personelin aldığı maaşların listesini veren sorgu. Bu sorguda 30 personel olmasına rağmen 29 personele ait maaş bilgisi listelenmiştir. Daha önce de bunun özellikle yapıldığı belirtilmişti. Birleştirme işlemleri yapılırken karşımıza şöyle bir problem çıkmaktadır. Birleştirme yapılan tablolardan ikinci tabloda birinci tablodaki her kaydın karşılığı olmazsa, karşılığı olmayan kayıtlar sorgu sonucunda sadece olmayan kayıtlar değil bilakis hiç kayıt gelmez. Bunun için "dış birleştirme" kullanılır. Dış birleştirme işlemi, kayıtları eksik olan tablonun şart tarafına "(+)" işareti konularak yapılır. Örnek 4 de buna ait uygulama yapılmıştır.



Şekil 11.2.3. Örnek Join Sorgu Ekranı-3

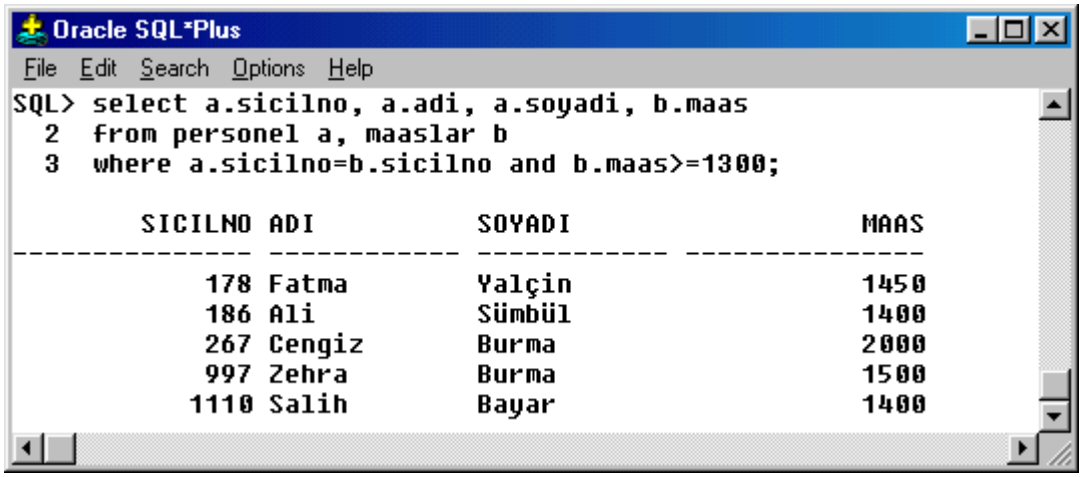
Örnek-4) Tüm personelin aldığı maaşların listesini veren sorgu. Maaşlar tablosunda 1077 sicil numaralı Orhan Sezgin isimli personele maaş bilgisi yazılması unutulmuştur. Buradan SQL'in aynı zamanda bir kontrol mekanizması olarak da kullanılabileceğini çıkarabiliriz.




```
SQL> select a.sicilno, a.adi, a.soyadi, b.maas
  2  from personel a, maaslar b
  3  where b.sicilno(+)=a.sicilno;

        SICILNO ADI        SOYADI                    MAAS
--------------- ---------- ---------- ---------------
            117 Fikret     Tatar                    800
            153 Orhan      Duru                    1100
            178 Fatma      Yalçin                  1450
            186 Ali        Sümbül                  1400
            245 Fatma      Papatya                 1050
            267 Cengiz     Burma                   2000
            326 Halil      Kangal                   795
            345 Serpil     Yücel                    800
            393 Saliha     Talay                    950
            456 Galip      Aslan                    975
            520 Nuran      Gölbakay                1150
            522 Selim      Arman                   1100
            590 Özgür      Irmak                   1000
            648 Ali        Kantar                  1250
            654 Refik      Erel                     825
            680 Hülya      Bakar                   1250
            772 Nihat      Ünalp                    825
            835 Kazim      Akbele                  1000
            849 Necla      Alakoç                   775
            890 Hüseyin    Kantar                   815
            898 Mehmet     Gezgin                   875
            960 Özgen      Öney                    1200
            967 Fatma      Erdem                    750
            997 Zehra      Burma                   1500
           1002 Zehra      Yüksek                   850
           1077 Orhan      Sezgin
           1110 Salih      Bayar                   1400
           1121 Nazan      Ulusoy                   825
           1240 Durmus     Bozkara                  950
           1290 Kaan       Yalçin                   875

30 rows selected.
```

Şekil 11.2.4. Örnek Join Sorgu Ekranı-4

Örnek-5) Aylık ücreti 1300 den büyük olan personelin listesini veren sorgu.



Şekil 11.2.5. Örnek Join Sorgu Ekranı-5

NOT : Bu bölüm sonuna kadar temel SQL komutları anlatılmıştır. SQL komutları çok kullanışlı ve yararlıdır. Örneğin: personel bilgileri içerisinden maaşı 1500 den büyük olanların listesi ve sayısı alınırken SQL dili değil de klasik programlama dili kullanılmış olsaydı böyle bir liste hangi programlama mantığı ile elde edilebilirdi. Sonuca ulaşabilmek için birçok satırdan meydana gelen kod yazılması gerekirdi. Programlama dilleri ile elde edilecek sonuçta izlenebilecek yöntemlerden biri şu olabilirdi : Bilgilerin yazıldığı dosyanın ilk kaydına gidilir, sonra dosyanın sonuna gelinene kadar her bir personel maaşının 1500'den büyük olup olmadığı şartı sorulur ve bu şarta uyan elemanlar listelenir. Kaç kişi olduğunun bilgisini bulmak içinde bu bilgi bir değişkende tutularak değişkenin değeri 1'er artırılarak şarta uyan kaç eleman olduğu bilgisine ulaşılır.

Aynı şekilde; her bölümde çalışan personel sayısı nedir veya kaç farklı bölüm vardır? SQL sorgusu için düşünülecek olursa bu tip sonuçların klasik programlama mantığı ile yapılabilmesinin ne kadar zor ve vakit alıcı olduğunu düşünün.

COUNT, AVG, SUM, DISTINCT, GROUP BY ve HAVING ile elde edilen sorgu sonuçlarının klasik programlama mantığı ile nasıl yapılabileceğini düşünün.

Bu örneklerden sonra SQL ile işlem yapmanın ne kadar kolay olduğu, SQL'in ne kadar yetenekli bir dil olduğu ve SQL'in önemi anlaşılabilir. Bu nedenle V.Basic, Delphi, Asp, PHP gibi programlama dillerinde SQL desteğini kullanarak işlemlerinizi kolaylaştırınız.